

COCC Continuing Education Apprenti Program Guide

Apprenti Education Provider

COCC Continuing Education is an educational provider – also known as “training provider” for the Central Oregon Apprenti technology apprenticeship program.

Apprenti provides a proven, reliable pipeline for underrepresented groups such as minorities, women and veterans to gain training, certification and placement within the talent-hungry tech industry. – <https://apprenticareers.org/about/>

Currently, training is offered at no cost to apprentices.

ADMISSION TO APPRENTI PROGRAM:

In order to begin the Apprenti Software Developer Program, a student must be successfully accepted into the program. Application and admission process requirements are listed below.

Potential Apprenti Program students must apply at <https://apprenticareers.org/locations/central-oregon/>

Apprenti Assessment is taken online and is divided into three sections with multiple choice answers: Math (60 questions), Logic & Critical Thinking (40 questions) and Soft Skills (14 questions) Potential students can choose which section of the assessment to start with, once a section is started, it must be finished – progress cannot be saved to be finished later. All sections do not have to be completed at once. Logic and critical thinking and softskills sections of the test take approximately 30 minutes each. The math section is longer, so 1.5-2.5 hours should be allocated. All three sections must be completed within 10 days. An option will be given to re-take the test if improved performance is desired. The assessment is not proctored, but must be completed individually without any assistance or support resources. A calculator may be used for the math section. Preparation resources are available.

Upon completion of the assessment, an email will be sent with the score of the three assessment areas along with information about comparisons with others who have completed the assessment (ranking).

Ranked Candidates: Those who perform well on the assessment will join the “ranked candidate pool.” These are the candidates who performed best on the assessment. For those in the ranked pool, beginning at the top of the list, candidates will be invited for interviews each time a group of companies present with jobs to fill. This will happen no less than once per quarter.

Un-ranked Candidates: Test re-takes are possible for up to three times, although there is a waiting period between tries.

Ranked Candidates who have a successful interview process and are offered an apprenticeship with a company then become part of the co-hort for classroom instruction, at which time a classroom schedule including all four programs within the Software Developer training area are announced.

CREDIT FOR PRIOR LEARNING

During Apprenti’s application process, Student Apprentices are able to apply for credit for prior learning. The Apprenti Program Manager will review the applicant’s prior learning experience and make a recommendation to the Oregon Apprenticeship Committee on whether or not to grant credit. The Oregon Apprenticeship Committee will evaluate and vote on whether to ratify the Apprenti Program Manager’s recommendation.

The guidelines follow:

- Prior learning credit will be granted on a like-for-like basis.
- Certifications considered for credit must be current and active. Expired certifications are not eligible for credit.
- Existing certifications must be the same certifications required by the apprenticeship curriculum.
- In occupations where certifications are not available, such as in software application development or web application development, an assessment test may be administered by the training provider to determine course placement.

Prior experience in a subject area is not guaranteed to be sufficient for granting credit, as apprentices are expected to have the same base level of skill and experience, so prior learning is only possible for an exact match between an industry certification and one required by the employer.

CODE OF CONDUCT – APPRENTI SOFTWARE DEVELOPER COHORT

This code of conduct outlines the expectations for participants within the COCC Continuing Education Apprenti community, as well as steps for reporting unacceptable behavior. COCC Continuing Education is committed to providing a welcoming and inspiring community for all and expect this code of conduct to be honored.

This Code of Conduct is in support of and in alignment with the broader reaching [COCC Student Rights and Responsibilities](#) as well as the Apprenti Apprentice Handbook.

Violations of this code of conduct will be addressed directly by the Lead Instructor. Students in violation of this code may be referred to the college's student conduct officer for investigation and review.

Principles of Community

Our community of staff, instructors, and current & former students strive to be kind. This is the guiding principle for conducting oneself within the Apprenti educational community.

While kindness can be generally understood, we recognize that kindness in a professional setting means to (but is not limited to):

- **Be friendly and patient**
- **Be welcoming:** We strive to be a community who welcomes and supports people of all backgrounds and identities. This includes, but is not limited to, members of any race, ethnicity, culture, national origin, color, immigration status, social and economic class, educational level, sex, sexual orientation, gender identity and expression, age, size, family status, political belief, religion, and mental and physical ability.
- **Be considerate:** Your work will be used by other people and you in turn will depend on the work of others. Any decision you take will affect users and colleagues and you should take those consequences into account when making decisions. Remember that we're a worldwide community, so you might not be communicating in someone else's primary language.
- **Give attribution, engage honestly:** Since our code often depends on the work and assistance of others, we will be sure to give proper attribution. As members of the community, we agree to represent ourselves truthfully, claim only work that is our own, properly attribute collaborations, and engage honestly in all assignments.
- **Be respectful:** Not all of us will agree all of the time, but disagreement is no excuse for poor behavior and poor manners. We might all experience some frustration now and then, but we cannot allow that frustration to turn into a personal attack. It's important to remember that a

community where people feel uncomfortable or threatened is not a productive one. Your presence may affect other's ability to learn.

- **Be careful in the words that we choose:** We are a community of learning professionals and we conduct ourselves professionally. Be kind to others both within the community and beyond the community. Do not insult, minimize, marginalize, or put down other participants, cultures, or sub-cultures. Contribute positively to the learning environment.
- **Try to understand why we disagree:** Disagreements, both social and technical, happen all the time. It is important that we resolve disagreements and differing views constructively. Remember that we're different. The strength of our community comes from its diversity—people from a wide range of backgrounds. Different people have different perspectives on issues. Being unable to understand why someone holds a viewpoint doesn't mean that they're wrong. Don't forget that it is human to err and blaming each other doesn't get us anywhere. Instead, focus on helping to resolve issues and learning from mistakes.
- **Harassment and exclusionary behavior aren't acceptable.**

Plagiarism

Plagiarism is the re-use of someone else's code, without the permission or license of the author. Claiming someone else's work as your own is disrespectful to the author and to your learning process. "Cargo culting," or utilizing the techniques of another without applying your own thinking, is also considered plagiarism.

Violating the terms of a license in a professional software development setting can result in lawsuits and endanger businesses. Doing so in the classroom can result in loss of assignment points, bad references from instructors, removal from all Code Fellows courses, and community bans. Consequences will depend on severity and are at the discretion of the conduct review panel.

Attribution

We encourage an open-source work ethic, and we solve problems collaboratively. The difference between properly attributed work and plagiarized submissions is found in citing source material and permission of the original author.

Proper attribution includes a description of what code was used or what help was given. A link back to the original source material or ideas should be included whenever possible. This belongs in a section titled "Attributions" of the README file for the relevant code.

Please note: an online code repository that has no license is presumed to be closed source and should not be used as a reference. Please include a license file in your own publicly published code.

COCC Academic Integrity Policy

Academic integrity shall encompass all core values and principles of honesty on and off campus at COCC and students are expected to complete their work ethically and honestly regardless of the **mode, delivery or type of instruction**. Our academic integrity policy prohibits all forms of cheating, plagiarism, unauthorized submissions, falsifications, improper collaboration, submitting identical work in courses without prior permission from the instructor, and attempting or assisting with others in violations of academic integrity would be a violation of students' rights and privileges. Furthermore, students should not use other classmates' work, or discussions without their expressed permission. In addition, no materials, content, statements, discussions, assignments or assessments, created by the instructor, can be copied, reused or distributed in any fashion without the written permission of the instructor. The penalties for violations of the academic Integrity policy could include grade reductions or course failure, depending of the severity of the penalty.

COCC Americans with Disabilities Statement

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should know of, or who need special arrangements in the event of evacuation, should make an appointment with the instructor as early as possible, no later than the first week of the term. Students may also wish to contact the COCC Disability Services Office in the Boyle Education Center, (541) 383-7583.

COCC Non-Discrimination Policy

Central Oregon Community College is an affirmative action, equal opportunity institution. It is the policy of the Central Oregon Community College Board of Directors that there will be no discrimination or harassment on the basis of age, disability, gender, marital status, national origin, race, religion, sexual orientation, or veteran status in any educational programs, activities or employment. Persons having questions about equal opportunity and non-discrimination, please contact Human Resources for referral to the appropriate personnel, 541-383-7236.

Harassment

We share a common understanding of what constitutes harassment, as it applies to a professional setting. Although this list cannot be exhaustive, we explicitly honor diversity in age, gender, gender identity or expression, culture, ethnicity, language, national origin, political beliefs, profession, race, religion, sexual orientation, socioeconomic status, and technical ability. We will not tolerate discrimination based on any of the protected characteristics above, including participants with disabilities.

Harassment includes, but is not limited to:

- Offensive comments (or "jokes") related to gender, gender identity and expression, sexual orientation, disability, mental illness, neuro-typicality, physical appearance, body size, race, age, regional discrimination, political or religious affiliation
- Unwelcome comments regarding a person's lifestyle choices and practices, including those related to food, health, parenting, drugs, and employment
- Deliberate misgendering. This includes deadnaming or persistently using a pronoun that does not correctly reflect a person's gender identity. You must address people by the name they give you when not addressing them by their username or handle.
- Physical contact and simulated physical contact (e.g., textual descriptions like "hug" or "backrub") without consent or after a request to stop
- Threats of violence, both physical and psychological
- Incitement of violence towards any individual, including encouraging a person to engage in self-harm
- Deliberate intimidation
- Stalking or following
- Harassing photography or recording, including logging online activity for harassment purposes
- Sustained disruption of discussion
- Unwelcome sexual attention, including gratuitous or off-topic sexual images or behavior
- Pattern of inappropriate social contact, such as requesting/assuming inappropriate levels of intimacy with others
- Continued one-on-one communication after requests to cease
- Deliberate "outing" of any aspect of a person's identity without their consent, except as necessary to protect others from intentional abuse
- Publication of non-harassing private communication

We encourage everyone to participate and are committed to building a community for all. Although we will fail at times, we seek to treat everyone both as fairly and equally as possible. Whenever a participant has made a mistake, we expect them to take responsibility for it. If someone has been harmed or offended, it is our responsibility to listen carefully and respectfully, and to do our best to right the wrong.

Examples

We don't require students to participate in formal conduct training. Rather, we rely on common sense and a commitment to follow these guidelines. To provide greater clarity, here are some examples of how this should be interpreted:

- A high-five is most welcome, as both parties are opting in. Avoid surprise hugs, back rubs, and general one-directional physical contact.
- If someone solicits feedback, focus on the material. Do not comment on physical appearance.
- In a learning environment, you'll find you have knowledge or insight someone else has not yet attained. Do not make others feel bad for not yet learning what you may find obvious. Avoid statements like "You haven't heard of Foo library?!?!" or, "Uh, obviously, you should just concat the strings."
- Our community constitutes people from a wide array of backgrounds. This is a great strength. Talk to people about their backgrounds and histories as a learner and a listener. "Tell me more about that," is a better option than, "Well, that's not what I experienced."
- Copying code from Stack Overflow, blogs, Google searches, online tutorials, etc., and pasting it into your own project is considered plagiarism. If you'd like to use someone else's code, at least retype the relevant portions yourself, and include a link back to the original in your project's README file.

Reporting Issues

Our community prioritizes marginalized people's safety over privileged people's comfort. We will not act on complaints regarding:

- 'Reverse' -isms, including 'reverse racism,' 'reverse sexism,' and 'cisphobia'
- Reasonable communication of boundaries, such as "leave me alone," "go away," or "I'm not discussing this with you"
- Refusal to explain or debate social justice concepts
- Communicating in a 'tone' you don't find congenial
- Criticizing racist, sexist, cissexist, or otherwise oppressive behavior or assumptions
- Diversity Statement

If you experience or witness unacceptable behavior—or have any other concerns—please report it by contacting us via rknox@cocc.edu. All reports will be handled with discretion. In your report please include:

- Your contact information.
- Names (real, nicknames, or pseudonyms) of any individuals involved. If there are additional witnesses, please include them as well.
- Your account of what occurred and if you believe the incident is ongoing. If there is a publicly available record (e.g. Slack channel, a mailing list archive or a public IRC logger), please include a link, reference, or screenshot.
- Any additional information that may be helpful.

After filing a report, a COCC representative will contact you personally, review the incident, follow up with any additional questions, and make a decision as to how to respond. The response team has broad

latitude to remove an offending student from class, or offer a warning, depending on the severity of the offense. No more than 1 warning will be given; a second offense will result in release from class, a refund according to the student contract, and a community ban.

If the person who is harassing you is part of the response team, they will recuse themselves from handling your incident. If the complaint originates from a member of the response team, a different member of the response team will handle it. We will respect confidentiality requests for the purpose of protecting victims of abuse.

Attribution & Acknowledgements

We all stand on the shoulders of giants across many open source communities. We'd like to thank the communities and projects that established codes of conduct and diversity statements as our inspiration, including Code Fellows, Todo Group, Python, Contributor Covenant, Geek Feminism, and Citizen Code of Conduct.

TERMINATION & DISMISSAL:

The skills training period of the Apprenti program is part of the student apprentice's initial probationary period. Therefore, registered apprenticeship rules apply to student apprentice termination from the program.

INITIAL PROBATIONARY PERIOD:

1. All apprentices are subject to an initial probationary period, stated in hours of employment during this time; an apprenticeship agreement may be terminated without cause. It is the period following the effective date of the apprentice's current registration into the program and during which the apprentice's appeal rights are restricted. (See ORS 660.126 (1g))
2. The initial probationary period must be reasonable in relationship to the full term of the apprenticeship unless otherwise required by Civil Service, CBA or law. It cannot exceed one year (12 months) or 25 percent of the length of the program, whichever is shorter. (See ORS 660.126 (1g))
3. During the initial probationary period, either party to the agreement may terminate the apprenticeship agreement upon written notice to the Apprenticeship and Training Division of the Oregon Bureau of Labor and Industries. (See ORS 660.126 (1g) & ORS 660.060 (6))
4. An appeal process is available to apprentices who have completed the initial probationary period. (See ORS 660.060 (6) & (7) and section X of this standard)

The probationary period shall be the first 400 hours of OJT hours of employment, or one year after the current registration to this standard, whichever is shorter. (See ORS 660.126 (g))

READMISSION:

Apprentices terminated from the Apprenti Oregon Program are eligible to re-apply. Circumstances of the termination will be considered by the Joint Apprenticeship and Training Committee (JATC) when determining eligibility for re-application.

The training provider (COCC) provides information only to the committee regarding termination and readmission decisions in accordance with the Family Educational Rights and Privacy Act (FERPA).

APPRENTI SAP (PROGRAM STANDARDS):

Apprenti Software Developer Program Standards:

- 1) Web Development Foundations NCTC Program
- 2) Full Stack Intermediate Software Developer NCTC Program
- 3) Advanced Software Development: JavaScript NCTC Program
- 4) Advanced Software Development Advanced JavaScript NCTC Program

Students in the four (4) above NCTC programs will adhere to the following academic policy:

Attendance:

Daily attendance is recorded for each student in each course. To pass a course 90% attendance is required, tracked and included as part of the overall grade. This policy is presented to students at the start of each course. If a student slips to 95% attendance, a warning is provided. When lack of attendance causes a student's grade to fall below 90% irrecoverably, the student is dismissed with a failing grade. In the event of unavoidable absences for reasons such as sickness or bereavement, the administrator and instructor may make case-by-case exceptions to this requirement, as long as coursework is made up in a timely manner with a passing grade, as described below.

Make-up Work:

All assignments missed due to absence must be completed and turned in to the instructor within five (5) business days of the student returning to class, unless the student in question makes explicit arrangements with the instructor. The instruction team will grade and return those assignments within five (5) business days.

Tardiness:

Attendance is significantly important, not only for the students as individuals but also for the class as a whole. Instructors have broad latitude to take whatever actions are necessary in order to correct tardiness problems with students. Tardiness is defined as arriving any later than the scheduled class time. Five (5) instances of tardiness will be tracked as the equivalent of a single day of missed attendance, and class attendance is tracked as part of each student's grade. 90% attendance is required for students to pass, so tardiness can lead to course failure. For example, if a course consists of 180 clock hours, no more than 18 hours are allowed for absences.

Grading Policy:

Students will receive a pass or no-pass grade on the official COCC non credit transcript, indicated by a P for pass or NP for no pass. In order to monitor student progress throughout each course and program, a point-based grading system is used. This system is introduced to the students on the first day of each program.

Students receive a specified number of points for assignments, quizzes, projects, participation, and other deliverables during the program. Program instructors maintain and update point totals throughout the program such that students can log in to the Learning Management System (LMS) software, Canvas by Instructure, and view their scores.

Instructors do not assign grades on any curve or adjusted system; all points are assigned completely based on completion of assignments and quality of work and energy invested. Students who maintain point totals of less than 90% of their potential points are considered to be underperforming.

Incomplete Grades:

If an assignment or set of assignments is missed, the student is allowed five (5) business days to make up missed work and submit missed deliverables. If a student does not believe that he/she can complete the missed work in this amount of time, he/she is responsible for establishing an alternative plan for making up the work with the course instructor(s).

Instructors verify grades for the official transcript within 10 business days of the end of each course or program.

Completion Requirements:

All four Apprenti Software Developer Programs require a final grade of 90% or higher and a minimum class attendance rate of 90% for completion. Students who successfully complete each of the four programs will receive a corresponding Non-credit Training Certificate.

WARNING, PROBATION, SUSPENSION:

Instructors meet with students on a regularly scheduled basis to assess progress, no less frequent than once per week. At each meeting, students are apprised of their successes and failures with opportunity to remediate and make up failing work. The instructor will assess and document progress, reporting to the student, the Continuing Education Program Manager and the Apprenti Program Manager to develop an improvement plan with specific dates for remediation. Students have at least two opportunities to re-take exams following each section before being terminated academically. The final determination decision is made by the JATC based on information provided by the instructor and the Continuing Education Program Manager.

Apprenti Software Developer Program Information

Program Name: Web Development Foundations

Program Award: Non-Credit Training Certificate

Number of Clock Hours: 160

Certificate As Awarded on Transcript: Web Development Foundations

PROGRAM DESCRIPTION:

The Web Development Foundations program prepares individuals for employment in the web development industry and is the first required module for the Apprenti Software Developer program. Web developers design and create websites. They are responsible for the look of the site and for the site's technical aspects, such as its performance and capacity. In addition, web developers may create content for the site.

Courses are offered in a face-to-face classroom setting, five days per week. Concepts are taught in a stacked module format, where a new concept is introduced in each class session, building upon what came before it. This is a challenging style that requires persistence, practice and collaboration, but allows more concepts to be introduced over the length of the course. This method helps students learn and retain more information in a short period of time.

The curriculum focuses on defining the structure of a web page, applying CSS and HTML to implement page layout styles, utilizing JavaScript to make web projects interactive and utilizing industry standard version control tools and workflows. Students will collaboratively design and create a single page web application using MVC architecture and will have the skills to move on to the next software development program, Advanced Software Development and Introduction to Python.

PROGRAM ENTRANCE REQUIREMENTS & PREREQUISITES:

Must be successfully accepted into the Apprenti Software Developer program.

COURSE(S):

CAPR201 Foundations of Software Development: Web Development Foundations **80 clock hours**

Build a strong software development foundation and learn how to use HTML, CSS, JavaScript and various libraries to create fully functional web apps. This course guides students toward developing a well-rounded foundation of skills necessary for modern web development. These skills include HTML, CSS and JavaScript coding; utilizing Git workflow processes; project organization; designing with wireframing; and employing introductory Agile development methods.

Course Outcomes:

1. Define the structure of a web page utilizing the semantic hierarchical structural conventions of HTML5
2. Apply CSS and HTML to implement page layout styles including grid, fluid and responsive techniques; and to style content with color, topography and images
3. Utilize JavaScript to make web projects interactive, such that user input is stored and processed to create updated and personalized content when a user interacts with a page.
4. Write JavaScript that leverages the fundamentals of Computer Science, the Object-Oriented Programming paradigm, basic data types, data structures and basic algorithms, so that the code is efficient, error-free and matches commonly accepted standards and practices of syntax and styles, as measured by the code's functionality and ability to pass a code linter

- Utilize industry standard version control tools and workflows, including Git and GitHub, to write functional, properly styled code and to work efficiently and harmoniously in professional environments

CAPR301 Foundations of Software Development: Intermediate Software Development **80 clock hours**

Becoming a well-rounded developer is much more than learning language syntax. In this intensive course, you will study the common core of software development, including MVC architecture, object-oriented and functional programming, and computer science fundamentals such as basic data structures and algorithms. Come learn how to create and launch web apps in HTML, CSS, and JavaScript, with the help of third-party APIs and libraries around the web.

Course Outcomes:

- Collaboratively design and create a single page web application from scratch using MVC architecture built with clean HTML, CSS and JavaScript that satisfies stakeholder requirements captured in user stories.
- Have the skills to move on to CAPR401 Advanced Software Development: Introduction to Coding in Python by successfully completing all requirements of the Web Development Foundations Program

VOCATIONAL OBJECTIVE:

Upon completion of the Non-credit Training Certificate in Web Development Foundations, the successful candidate will have the skills necessary to attain an entry-level website development job or internship. By completing the requirements and receiving the Non-credit Training Certificate in Web Development Foundations award, the successful candidate will have completed the pre-requisites required to enter the next module of the Apprenti Software Developer training area: Advanced Software Development and Introduction to Python Program.

PROGRAM STANDARDS:

Students in the Web Development Foundations Program will receive a Pass/No-Pass Grade on the official COCC Non-credit Transcript, based on the following course assessments:

- Attendance and participation: Professional code of conduct and adherence to attendance policy. No more than 10% of face-to-face classes can be missed
- Homework: required with passing grade of 90% or above
- Projects for portfolio
- Labs with code-writing challenges
- Minimum of 75% grade upon completion of coursework
- Quizzes and tests
- Peer programming, co-lab (co-working) team projects

ACADEMIC PROGRESS RECORDS:

Students in the Web Development Foundations Program will receive weekly course status surveys and a bi-weekly progress report consisting of attendance and performance based on measured outcomes.

PROGRAM CALENDAR:

The Web Development Foundations Program will be offered Summer 2018 and in subsequent terms as the Apprenti Software Developer partnership program demands.

Web Development Foundations Program – 160 Clock Hours

June 11-July 6; 8 am to 5 pm each day with 1 hour lunch period; M-F with this exception: class to be held June 30 in lieu of July 4 holiday

Program Name: Full Stack Intermediate Software Developer

Program Award: Non-Credit Training Certificate

Number of Clock Hours: 160

Certificate As Awarded on Transcript: Full Stack Int Software Dev

PROGRAM DESCRIPTION:

The Full Stack Intermediate Software Developer program prepares individuals for employment as a software developer. Studying the common core of software development, including MVC architecture, object-oriented and functional programming, students will gain a solid understanding of intermediate software development concepts. This is the second required module for the Apprenti Software Developer program. Students will create and launch web apps in HTML, CSS and JavaScript with the help of third-party APIs and libraries from around the web. Students will study professional software development techniques and practices while advancing skills.

Courses are offered in a face-to-face classroom setting, five days per week. Concepts are taught in a stacked module format, where a new concept is introduced in each class session, building upon what came before it. This is a challenging style that requires persistence, practice and collaboration, but allows more concepts to be introduced over the length of the course. This method helps students learn and retain more information in a short period of time.

The curriculum focuses on programming and computer science fundamentals such as basic data structures and algorithms. Students will collaboratively design and create a single page web application using MVC architecture and will have the skills to continue advancing skills or obtain an entry-level web development role or internship.

PROGRAM ENTRANCE REQUIREMENTS & PREREQUISITES:

Successful completion of Web Development Foundations Non-credit Training Certificate.

COURSE(S):

CAPR311 Full Stack Intermediate Software Developer
160 clock hours

In this intensive course, students will study the common core of software development, including MVC architecture, object-oriented and functional programming, and computer science fundamentals such as basic data structures and algorithms. Students will create and launch web apps in HTML, CSS and JavaScript with the help of third-party APIs and libraries from around the web. Upon successful completion of this 160-hour course, students will have a solid understanding of intermediate software development concepts and be prepared to either continue advancing skills or obtain an entry-level web development role or internship.

Course Outcomes:

1. Students will collaboratively design and create a single page web application from scratch using MVC architecture built with clean HTML, CSS and JavaScript that satisfies stakeholder requirements captured in user stories.
2. Students will construct the view layer of a basic MVC web application using jQuery, JavaScript and templates.
3. Students will create a model layer using Node.js, PostgreSQL
4. Students will create a full-stack application using Agile processes
5. Students will perform a production deployment of a web-application to Heroku

VOCATIONAL OBJECTIVE:

Upon completion of the Non-credit Training Certificate in Full Stack Intermediate Software Developer, the successful candidate will have the skills necessary to attain an entry-level website development position or internship and will have completed the pre-requisites required to enter the next module of the Apprenti Software Developer training area: Advanced Software Development: JavaScript.

PROGRAM STANDARDS:

Students in the Full Stack Intermediate Software Developer Program will receive a Pass/No-Pass Grade on the official COCC Non-credit Transcript, based on the following course assessments:

1. Attendance and participation: Professional code of conduct and adherence to attendance policy.
No more than 10% of face-to-face classes can be missed
2. Homework: required with passing grade of 90% or above
3. Projects for portfolio
4. Labs with code-writing challenges
5. Minimum of 75% grade upon completion of coursework
6. Quizzes and tests
7. Peer programming, co-lab (co-working) team projects

ACADEMIC PROGRESS RECORDS:

Students in the Full Stack Intermediate Software Developer Program will receive weekly course status surveys and a bi-weekly progress report consisting of attendance and performance based on measured outcomes.

PROGRAM CALENDAR:

The Full Stack Intermediate Software Developer program will be offered Summer 2018.

Full Stack Intermediate Software Developer - 160 Clock Hours
July 23-August 17; 8 am to 5 pm each day with 1 hour lunch period

Program Name: Advanced Software Development: JavaScript

Program Award: Non-Credit Training Certificate

Number of Clock Hours: 200

Certificate As Awarded on Transcript: Adv Sftware Dev: JavaScript

PROGRAM DESCRIPTION:

The Advanced Software Development: JavaScript program prepares individuals for employment in the software development industry and is the third of four required modules for the Apprenti Software Developer program. Software developers are the creative minds behind computer programs. Some development applications that allow people to do specific tasks on a computer or another device. Others develop the underlying systems that run the devices or that control networks. Software developers analyze users' needs and then design, test and develop software to meet those needs. They are responsible for the entire development process for a software program, identifying core functionality that users need from software programs. Software developers must also determine user requirements such as level of security and performance needs. Developers write code or give instructions to others to write code.

Courses are offered in a face-to-face classroom setting, five days per week. Concepts are taught in a stacked module format, where a new concept is introduced in each class session, building upon what came before it. This is a challenging style that requires persistence, practice and collaboration, but allows more concepts to be introduced over the length of the course. This method helps students learn and retain more information in a short period of time.

The curriculum focuses on Full-Stack JavaScript development, core competencies and best practices. Students will build back-end, server-side web applications with Node.js. Throughout this intensive course, students will study data structures and algorithms, professional software development techniques, and established industry best practices while advancing skills in full-stack JavaScript. Upon completion of Advanced Software Development: JavaScript, students will have the skills to move on to the next software development program, Advanced Software Development : Advanced JavaScript.

PROGRAM ENTRANCE REQUIREMENTS & PREREQUISITES:

Successful completion of the Full Stack Intermediate Software Developer Non-credit Training Certificate.

COURSE(S):**CAPR PR411 Advanced Software Development: JavaScript
200 clock hours**

Learn to write clean, well-tested, JavaScript code using industry standard software engineering patterns.

Course Outcomes:

1. Build and contribute to a server-side and client-side application demonstrating industry best practices, architecture/framework that promotes maintainability, scalability, and collaboration.
2. Apply Computer Science fundamentals in analyzing and choosing algorithms, differentiating between JavaScript coding patterns and practices, and using byte formats in application building.
3. Assess differences between Object Relational Mapping and Object Document Mapping.
4. Demonstrate the ability to model data using a NoSQL Object Document Mapper for MongoDB.
5. Host a full-stack web application on Heroku.

VOCATIONAL OBJECTIVE:

By completing the requirements and receiving the Non-credit Training Certificate in Advanced Software Development: JavaScript award, the successful candidate will have completed the pre-requisites required to enter the next module of the Apprenti Software Developer training area: Advanced Software Development: Advanced JavaScript Program.

PROGRAM STANDARDS:

Students in the Advanced Software Development: JavaScript Program will receive a Pass/No-Pass Grade on the official COCC Non-credit Transcript, based on the following course assessments:

1. Attendance and participation: Professional code of conduct and adherence to attendance policy.
No more than 10% of face-to-face classes can be missed
2. Homework: required with passing grade of 90% or above
3. Projects for portfolio
4. Labs with code-writing challenges
5. Minimum of 75% grade upon completion of coursework
6. Quizzes and tests
7. Peer programming, co-lab (co-working) team projects

ACADEMIC PROGRESS RECORDS:

Students in the Advanced Software Development: JavaScript Program will receive weekly course status surveys and a bi-weekly progress report consisting of attendance and performance based on measured outcomes.

PROGRAM CALENDAR:

The Advanced Software Development: JavaScript Program will be offered Fall 2018 and in subsequent terms as the Apprenti Software Developer partnership program demands.

Advanced Software Development: JavaScript - 200 Clock Hours
September 4-October 5; 8 am to 5 pm each day with 1 hour lunch period

Program Name: Advanced Software Development: Advanced JavaScript

Program Award: Non-Credit Training Certificate

Number of Clock Hours: 208

Certificate As Awarded on Transcript: Adv Sftware Dev Adv JavaScript

PROGRAM DESCRIPTION:

The Advanced Software Development: Advanced JavaScript program prepares individuals for employment in the software development industry and is the final required module for the Apprenti Software Developer program. Software developers are the creative minds behind computer programs. Some development applications that allow people to do specific tasks on a computer or another device. Others develop the underlying systems that run the devices or that control networks. Software developers analyze users' needs and then design, test and develop software to meet those needs. They are responsible for the entire development process for a software program, identifying core functionality that users need from software programs. Software developers must also determine user requirements such as level of security and performance needs. Developers write code or give instructions to others to write code.

Courses are offered in a face-to-face classroom setting, five days per week. Concepts are taught in a stacked module format, where a new concept is introduced in each class session, building upon what came before it. This is a challenging style that requires persistence, practice and collaboration, but allows more concepts to be introduced over the length of the course. This method helps students learn and retain more information in a short period of time.

The curriculum focuses on fundamental computer science concepts, analyzing algorithms, testing asynchronous actions, communicating over the web in real time, utilizing cryptography concepts and creating online brand, network and resume. At the completion of this program, students will have the skills and portfolio to obtain an entry level position as a Full-Stack JavaScript or Front-End Web Developer.

PROGRAM ENTRANCE REQUIREMENTS & PREREQUISITES:

Successful completion of Advanced Software Development: JavaScript Non-credit Training Certificate.

COURSE(S):

CAPR PR412 Advanced Software Development: Advanced JavaScript 208 clock hours

Students will focus on client-side development centered around designing and building “Single Page Apps” using a modern JavaScript application framework or library that integrates with an application server.

Course Outcomes:

1. Apply Computer Science fundamentals in analyzing algorithms, testing asynchronous actions, and communicating over the web in real time using websockets.
2. Utilize cryptography concepts with public/private keys and digital certificates.
3. Create an online brand, network, and resume; and will demonstrate aptitude in personal and technical interviews of varying formats.

VOCATIONAL OBJECTIVE:

Upon completion of the Non-credit Training Certificate in Advanced Software Development: Advanced JavaScript, the successful candidate will have the skills necessary to attain an entry-level software developer job or apprenticeship. By completing the requirements and receiving the Non-credit Training Certificate in Advanced Software Development: Advanced JavaScript award, the successful candidate will have completed the pre-requisites required to begin an onsite year-long apprenticeship in the Apprenti Software Developer training area.

PROGRAM STANDARDS:

Students in the Advanced Software Development: Advanced JavaScript Program will receive a Pass/No-Pass Grade on the official COCC Non-credit Transcript, based on the following course assessments:

1. Attendance and participation: Professional code of conduct and adherence to attendance policy.
No more than 10% of face-to-face classes can be missed
2. Homework: required with passing grade of 90% or above
3. Projects for portfolio
4. Labs with code-writing challenges
5. Minimum of 75% grade upon completion of coursework
6. Quizzes and tests
7. Peer programming, co-lab (co-working) team projects

ACADEMIC PROGRESS RECORDS:

Students in the Advanced Software Development: Advanced JavaScript Program will receive weekly course status surveys and a bi-weekly progress report consisting of attendance and performance based on measured outcomes.

PROGRAM CALENDAR:

The Advanced Software Development: Advanced JavaScript Program will be offered Fall 2018 and in subsequent terms as the Apprenti Software Developer partnership program demands.

Advanced Software Development: Advanced JavaScript - 208 Clock Hours

October 8-November 14: 8 am to 5 pm each day with 1 hour lunch period; no class Monday November 12 due to college observed holiday

Software Developer Instructional Team

COCC Continuing Education (training provider)

RACHEL KNOX**Program Manager Continuing Education**

B.A. in English/British Literature, Oakland University. At COCC since 2003.

NICHOLAS CARIGNAN**Lead Instructor**

Certificate in Advanced JavaScript, Code Fellows, Seattle. At COCC since 2018.

ADAM DUQUETTE**Lead Instructor**

B.S. in Applied Computer Science, Oregon State University-Cascades, Bend. At COCC since 2018.

TIM LEAVEY**Teaching Assistant**

B.A. in Asian Studies and Japanese, University of Oregon, Eugene. At COCC since 2018.